Exhibit 4

Exhibit 4

ellipsus SYSTEMS

White Paper - infiniteMAP™

Ellipsus’ Mobile Application Provisioning System

Heterogeneous environment, a new challenge

With the introduction of the mobile Internet, the communication landscape is permanently changing from a predominantly homogenous environment, where all user workstations generally had the same characteristics and capabilities, regardless of brand and type. Instead we now see a number of different ”workstations”, ranging from desktop devices to handheld devices, and networks ranging from fibre to wireless. Such a heterogeneous environment poses new challenges, but also possibilities for application developers, service providers and -most importantly- end users.

Because of their ubiquity, particularly in countries that do not have an extensive ”wired” landscape, a majority of the workstations accessing the Internet will have wireless connectivity, limited processing capabilities, limited memory and limited user interfaces. In short, they are thin clients. Because of the noted limitations, it becomes necessary to optimise the use of the device. One way to do that is by allowing it to become something more than just a browser. With a browser, we can view and interact with content. If it is possible to download software to the device in real-time, we can turn the device into an integrated part of an application instead. This is a very powerful and dynamic way to maximize resource utilization.

Executable Internet

With the application running on the client device, more responsive, interactive and dynamic applications can be realized. Data communication can be optimised for the current application, which prolongs battery life in handheld devices. Software version control is a non-issue since the client software can be updated next time it interacts with the server. The client update process is automated, which is extremely important in a mobile and wireless environment. The server version and client version is automatically kept in sync at all times. The concept of running the application, rather than a browser, on the client is called the executable Internet.

Provision applications

It becomes necessary to define mechanisms that deal with thin clients, their limited and intermittent availability of bandwidth, and new capabilities, such as context driven operation based on location, user preferences and device characteristics. The ability to provision applications to devices in real-time, and create revenues from it, is the mechanism for this.

Platform-neutral

Meanwhile, service providers, such as telcos and ISP’s, are looking for ways to deliver services, rather than communication time to their customers. They want a platform-neutral solution, that can be used to offer services to their subscribers.

Platform-neutrality means that the provider:

Plug and unplug can plug, and unplug services from the service portal as needed;

Flexible does not have to make an irrevocable commitment to any individual application provider platform;

Maximize can maximize the user experience by offering a rich portfolio of high quality services;

Billing can simultaneously and in parallel operate several billing models;and

Scale can scale up infrastructure to meet demand and ensure operational security.

Ellipsus’ Mobile Application Provisioning System - infiniteMAP™

infiniteMAP™, by Ellipsus, is leading the technology for provisioning of applications and objects to devices.

It is squarely positioned to meet the requirements by:

Rapid deployment

allowing the enterprise to rapidly deploy dynamic applications and objects to devices, and integrate with supporting business operations;,

Cross network and cross device

providing cross network provisioning of applications and objects, wired as well as wireless support to a wide variety of device types;,

Application neutral

being application neutral, supporting provisioning of objects for web-, WAP-, Java™ -, SMS and legacy applications;, and

Platform neutral

offering platform neutrality, allowing each application to execute on its application server, independent of other applications and the operator infrastructure.

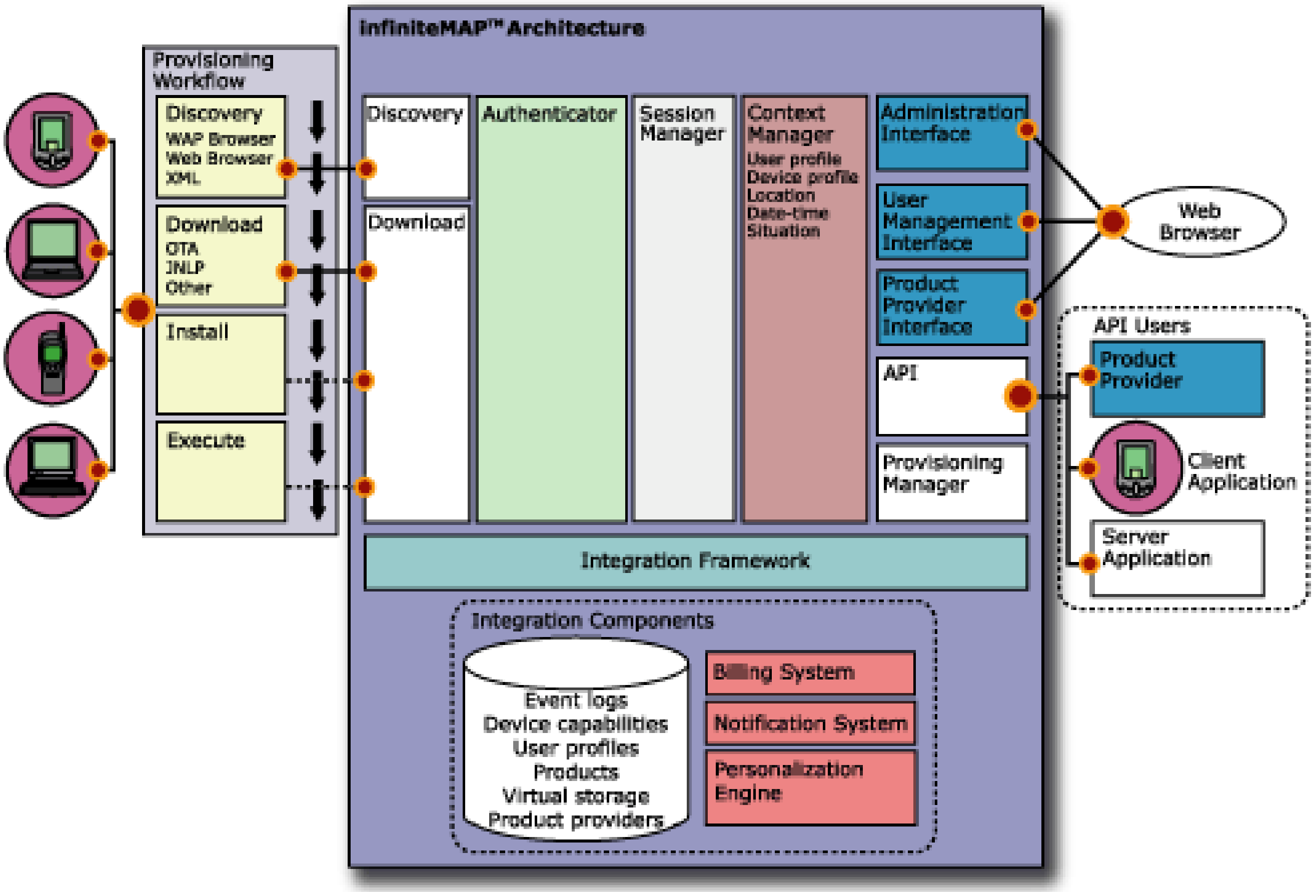
What is application provisioning?

In its simplest form, application provisioning can be described as the capability to receive a request for an object, finding a suitable version of the requested object and provide it to the requestor. This function is required since the user requesting the application can be using any one of a number of different client device types, each with its own characteristics, limitations and configuration. In order for a server application to be open and flexible, it will need to accommodate numerous such devices, and add new devices easily as they are introduced into the marketplace. infiniteMAP takes care of this issue and lets the application developer focus on the actual application domain. In practical operation, application provisioning is more complex than what the simple description above indicates.

”the capability to receive a request for an object, finding a suitable version of the requested object and provide it to the requestor”

infiniteMAP - vending machine

infiniteMAP can also be viewed as a virtual vending machine, where the customer can view a selection of products, make a selection, be presented with a cost, authorize payment and then use the selected product.



Clients devices

Various clients devices are handled by the mobile users, accessing infiniteMAP. The client device capabilities and user profile determines available services. Each client traverses a number of steps collectively called the provisioning workflow. The user may use a number of different provisioning applications to perform the workflow, such as WAP browsers, WEB browsers, or proprietary XML-schemes.

Authenticator

The Authenticator will ensure that each user is authenticated at all times and that a matching provisioning adapter is used, based on the currently used provisioning application.

Session manager

The Session manager maintains a user-centric session, which makes it possible for the user and device to switch between applications, while making sure the provisioning workflow is performed properly.

Context manager

The Context manager creates an XML-document based on user profile, device capabilities and other context, such as location and user situation (work, leisure). The user may subscribe to only certain topics from those made available, for example.

Administration interface

The Administration interface allows the operator to customize the system, approve deployed applications and perform customer care tasks

User manager interface

The User manager interface allows the user to customize his/her interaction with infiniteMAP. It is possible to create topics and thereby affect the presentation order in the service menu. Some users may prefer games listed first while others prefer financial applications, for example.

Product provider interface

The Product provider interface is a single point entry into the administrative system associated with infiniteMAP. This interface allows product providers to configure new content and services, device capability profiles and billing rules, specific to the application that is being published through infiniteMAP.

API

The Application Programming Interface consists of three voluntary interfaces, allowing client-side applications, server-side applications and product providers, to interact with infiniteMAP. This may include billing events, authentication, product deployment and updating virtual storage.

Provisioning manager

The Provisioning manager will fetch the requested object from contracted product providers. It will package it in a format suitable for the device type and transmit it to the device. It also contains an object cache to optimise performance.

Integration framework

The Integration framework provides the most flexible and efficient way to interface to existing Operational Support Systems (OSS) and Business Support Systems (BSS). It offers several integration points, which promote scalable and robust interfacing with all today known sub-systems, including billing systems.

User access to the vending machine

The following steps are performed by infiniteMAP
when a user purchases a ”product” such as the use of a game:

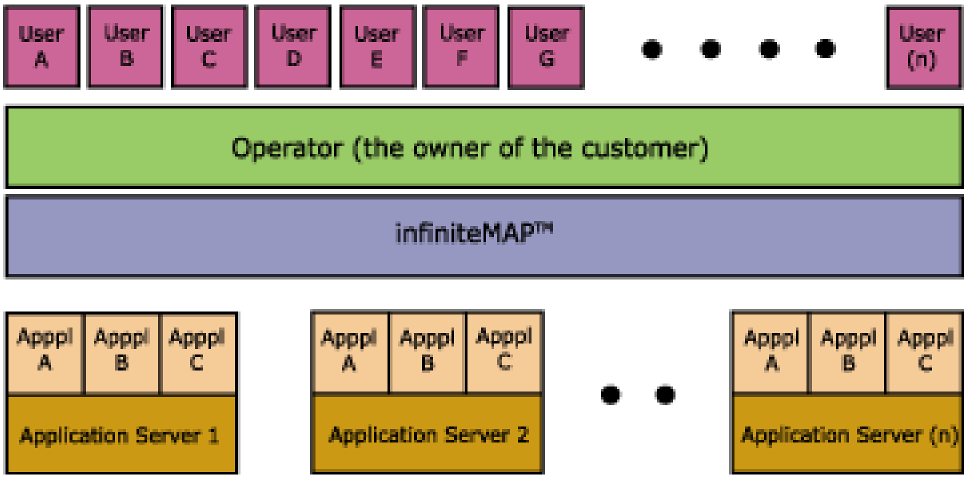
- investigating the user profile of the user who is requesting an application menu,
- providing a context driven portal based on user profile, location, device type, and other contex.
- receiving a request for an application,
- investigating what type of device the requestor is using
- investigating the characteristics, limitations and current state of the device,
- finding an object that matches the client device and user profile,
- fetching the object from an object storage,
- packaging the object in a format suitable for the device,
- adding billing control mechanisms to the object,
- sending it to the device,
- initiating billing for the delivery.

The benefits to the user is a personalized view of products based on user preferences, device type, most frequently accessed products, current location and other context. The user will be served with a selection of matching products that hide technical implications, allowing selection to be based solely on product functionaity.

Platform-neutral architecture

The application developer publishes applications through infiniteMAP, not on top of it. In other words, this is not yet another proprietary platform. Instead infiniteMAP offers platform neutrality, which means that application providers can develop and deploy on application servers of their choice, then configure and publish through it.

There are no requirements to conform to any specific application programming interface (API).

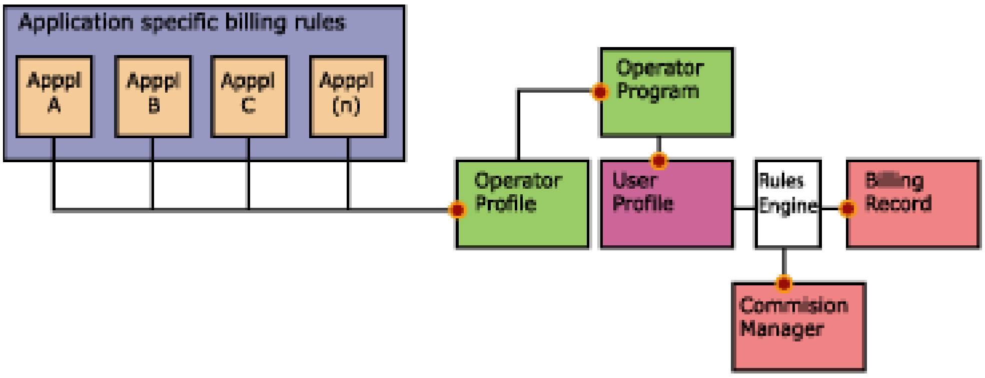


Services over one single interface

infiniteMAP makes the application servers transparent to the operator, and will therefore offer great flexibility in deployment of services while also create maximum user value by allowing each application to execute in its optimum environment.

Billing model

infiniteMAP provides an adaptive billing interface, making it quite easy to integrate with existing billing systems.



Flexibility in deployment

Each application is unique and its characteristics will inevitably affect the optimal billing model for it. So, for example, a scientific calculator may be downloadable a finite number of times after initial payment. A game of chess, on the other hand, might be billed per session, regardless of how many times the user transitions between devices.

Operator profile

Once the initial billing information is fetched from the application configuration record, it is passed to the operator profile as a ”candidate fee”. This module may reduce or increase the candidate fee, or affect it some other way depending upon operator-specific rules that may be applied.

User profile

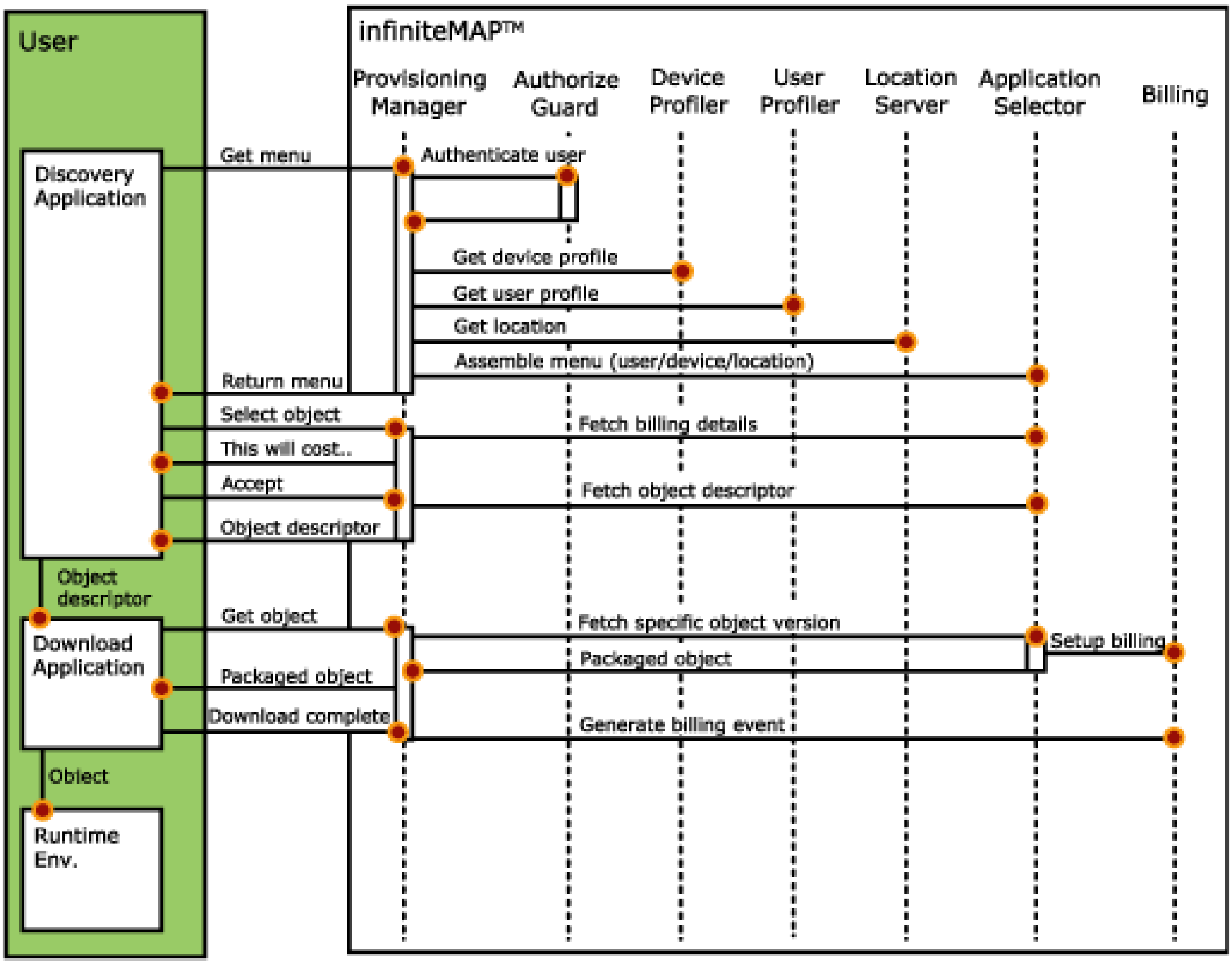
The user profile also affects what fee, if any, is charged for accessing a service. The user may have a subscription that comes with special privileges, such as a general discount. There might also be a frequent user program in place, managed by the operator program module, giving various discounts based on how frequent the user accesses services via infiniteMAP or frequent use of specific services.

Commission manager

The final billing record is communicated as a call data record (CDR) to the billing system. At that time the billing record is also communicated to the commission manager, which if appropriate, will divide the final charge between the owner of the customer, system operator and the application provider. The commission manager is rule based and can be set up as desired. When all three roles are performed by one party, the commission manager will have no effect.

Processing Workflow

The process of application discovery, selection, download and execution is described in the figure below.



Workflow for major events and interactions

Processing Workflow

The figure on the previous side only illustrates major events and interactions.

Discovery application

The user will use a discovery application, such as a web browser, a WAP browser or some proprietary mechanism, to connect to infiniteMAP. Upon connection, infiniteMAP will authenticate the user. The authenticator supports single-sign-on, which means that if the user is accessing infiniteMAP via a cell phone, authentication could be performed via the cellular network rather than requesting user name and password.

Once connected to infiniteMAP, the device profile, user profile and user location is fetched. Based on this information, a dynamic view (menu) is created and returned to the user.

Download application

The user can now browse the menu and choose an application or object for download. Once an object has been selected, infiniteMAP will fetch specific details and present these to the user. If the user accepts the charges (if any) that will be incurred, infiniteMAP returns an object descriptor.

The object descriptor is passed to the download application, which conforms to the MIDP Over The Air (OTA) specification, Java Network Launch Protocol specification (JNLP) or some other Internet download mechanism. The download application will use the object descriptor to request the object file from infiniteMAP.

Upon receiving the object request, infiniteMAP will package the object and set up object specific billing mechanisms. The packaged object is returned to the user.

Once the download is successfully completed, the download application may notify infiniteMAP. infiniteMAP completes the transaction by generating a billing event.

Runtime environment

The downloaded application hands over the object file to the runtime environment, which could be a J2ME platform, a J2SE platform, an applet sandbox or some other environment capable of accepting the object.

Conclusions

Being mobile means freedom of movement for the individual user, and requires the freedom to communicate when needed, not only when given a chance to. This means that the user must be able to communicate using a number of different devices, wired as well as wireless. The user should benefit from the mobile environment even with its inherent latencies in the network, limited bandwidth, sporadic connectivity, limited computing resources in handheld devices, and limited user interfaces. infiniteMAP is built for the mobile communication environment exclusively, and optimises these limitations.

This means specifically that infiniteMAP:

Best support

Has the best wireless and mobile support in the form of protocol gateway (WAP 1.2, 2.x), security protocols (WTLS, TLS), SMS integration and location server.

Application neutral

Is application neutral, which means that you can download midlets (J2ME), applets, Java applications (J2SE), native code applications, web URLs, WAP URLs, SMS messages and objects, such as ring tones, video clips a s o.

Cross network

Supports cross network discovery and download. This means that you can use web browsers (HTML), WAP browsers (WML), SMS and XML for application discovery. You can use OTA, Internet, Web Start (JNLP) for application download.

100% Java, supporting J2EE

It is developed in 100% Java, supporting J2EE, which allows us to provide premium support and allows the customer to deploy on the current platform.

Integration oriented design

The design is integration oriented, which means that we can efficiently integrate with existing legacy systems.

Context driven

Is context driven, which means that we consider user preferences, device profile, location, situation and other factors when interacting with the user.

About Ellipsus System

Ellipsus Systems develops, markets, and supports a suite of universal, platform neutral middleware products that dynamically provision and manage applications and content to a wide range of consumer mobile device types. Through its highly regarded mobile application provisioning technology and infinite suite of software products, Ellipsus enables mobile operators and enterprises to extend their reach to the mobile Internet by leveraging existing infrastructure with secure, scalable and open solutions. Founded in 1999, Ellipsus Systems, Inc. is a closely-held Delaware corporation with corporate and North American headquarters in Dallas, Texas, and sales and engineering offices in Sweden.

www.ellipsus.com • info@ellipsus.com • wap.ellipsus.com

Corporate Headquarters, Dallas: Ellipsus Systems Inc • P.O. Box 797525 • Dallas, TX 75379-7525, USA
Phone: +1-214-228-6252

Nordic Sales Office, Stockholm: Ellipsus Systems AB • Ringvägen 100 C • SE-118 60 Stockholm, Sweden
Phone: +46 (0)8-556 97 330 • Fax: +46 (0)8-556 97 339

Engineering Center, Växjö: Ellipsus Systems AB • Regementsgatan 9 • SE-352 36 Växjö, Sweden
Phone: +46 (0)470-73 71 60 • Fax: +46 (0)470-73 71 61